FAST NETWORK QUERYING ALGORITHM FOR SEARCHING LARGE-SCALE BIOLOGICAL NETWORKS

Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon

Department of Electrical & Computer Engineering
Texas A&M University, College Station, TX 77843-3128, USA
msahraeian@tamu.edu, bjyoon@ece.tamu.edu

ABSTRACT

Network querying aims to search a large network for subnetwork regions that are similar to a given query network. In this paper, we propose a novel algorithm for querying large scale protein interaction networks. In this algorithm, we iteratively compute the correspondence scores between nodes in the query and the target networks using semi-Markov random walk. Based on these scores, we reduce the search space in the target network by discarding irrelevant nodes. The scores are re-estimated in each iteration after removing such nodes, which ultimately leads to more accurate querying result. Numerical experiments based on both synthetic and real networks show that the algorithm can efficiently find accurate querying results.

Index Terms— Network querying, protein-protein interaction (PPI) network, semi-Markov random walk, network reduction

1. INTRODUCTION

Network biology quantitatively describes and analyzes biological systems in terms of the interactions among the constituent biological components [1]. The protein-protein interaction (PPI) network graphically represents all interactions between pairs of proteins in a given organism, and it provides an important resource for studying network biology. For example, comparative analysis of PPI networks of different species can reveal similar subnetwork regions in the respective networks. These regions may correspond to functional modules, such as signaling pathways or protein complexes that are conserved in the compared species.

Network querying aims to search a large network of an organism (the "target" network) to find subnetwork regions that look similar to a given query network of another organism. The query network can be a well-studied functional module in a model organism. Network querying allows us to predict similar modules in other less-studied organisms, thereby providing a way to transfer biological knowledge across different organisms [2].

When searching for matching network regions in different PPI networks, an effective network querying algorithm should incorporate the interaction pattern similarity (i.e., topological similarity) as well as the protein sequence similarity (i.e., node similarity) into the search process. However, with thousands of proteins in a typical PPI network, intertwined in a complicated manner, network querying is computationally very challenging. In fact, as a generalization of the subgraph isomorphism problem, it is NP-complete. Several network querying techniques have been proposed so far using some constraints to simplify the problem [3, 4, 5, 6, 7]. Most of these techniques are restricted to special type of queries [3, 4] or basically ignore the topology of the query network [5, 6, 7]. They also do not explicitly use the sequence similarity between proteins in query and target networks but just define some threshold and seek for matching nodes in the given networks whose similarity score exceeds the given threshold. Furthermore, these schemes generally have high computational cost, which increases exponentially with respect to the size of the query network.

In this paper, we propose a novel network querying algorithm that adopts an efficient network reduction approach based on node correspondence scores computed by semi-Markov random walk modeling. This modeling allows us to seamlessly combine topological similarity and sequence similarity to estimate the node correspondence scores. Based on these scores, we iteratively reduce the target network by removing nodes with minimal correspondence scores, to identify the subnetwork region that closely matches the query network. Re-estimating the scores based on the reduced target network in each iteration yields an improved set of correspondence scores that can ultimately lead to more accurate querying results. The computation complexity of the proposed algorithm is polynomial in terms of the size of both networks, and it can be efficiently implemented by exploiting the sparsity of typical protein interaction networks. To demonstrate the effectiveness of the proposed network querying algorithm, we evaluate the performance using both synthetic and real PPI networks. We show that our algorithm can accurately find matching network regions and it takes only around ten seconds to search for a general query network with ten or more nodes in a large target network with tens of thousands of nodes, using Matlab on a personal computer.

2. PROPOSED METHOD

2.1. Semi-Markov Random Walk

Let's represent the target PPI network as an undirected weighted graph \( G = (V, E, w) \) with a set \( V \) of \( n \) vertices, representing proteins, a set \( E \) of edges, representing interactions between proteins, and a weight function \( w : E \rightarrow \mathbb{R} \), representing interaction reliabilities. Let \( G_Q = (V_Q, E_Q, w_Q) \) represent the query network with \( k \) vertices. Also, let \( S_{ij} \) denote the sequence similarity score (e.g., BLAST score) between the query protein \( q_i \) in \( G_Q \) and the target protein \( v_j \) in \( G \). As illustrated in Fig. 1, we want to search for a set of vertices in \( G \), such that they have high sequence similarity with the query nodes and form a topologically similar subnetwork to the query network.

We define a correspondence score between each pair of vertices of the two networks by modeling the problem as a semi-Markov random walk. A semi-Markov process is a stochastic process that makes state transitions according to a Markov chain but spends a random amount of time between transitions. In this modeling, the walker
Fig. 1. Network querying problem. \( G_Q \) is the query network, and \( G \) is the target network. Solid lines represent the interactions between proteins of a given network, while dashed lines connect matching nodes with high topological and sequence similarity.

A network querying problem takes simultaneous random steps on both query and target networks, \( G_Q \) and \( G \). Therefore, the position of the walker is determined by a pair of vertices one from query and one from target network. The mean time that the random walker stays at a specific pair of vertices is modeled to be proportional to the sequence similarity between those pair; hence the walker will stay longer at vertices that are more similar. Based on this model, we compute the long-run proportion of time that the random walker stays at a given pair of vertices, which is used as the correspondence score between those vertices.

Thus, we construct a Markov chain for each of query and target networks so that each vertex in those graphs corresponds to a state in the Markov chain. For graph \( G \), we define the Markov chain as a directed graph with the same set of nodes \( V \), and the transition probability matrix \( A = [a_{ij}] \), where \( a_{ij} \) is the transition probability from states \( v_i \in V \) to state \( v_j \in V \), is defined based on the weight function \( w \) as:

\[
a_{ij} = \frac{w_{ij}}{\sum_{h \in N(v_i)} w_{ih}},
\]

where \( N(v_i) \) is the set of neighbors of node \( v_i \). In a similar manner, we define a Markov chain on the query network \( G_Q \) with transition probability matrix \( A_Q \).

Assume that we perform a simultaneous random walk on \( G_Q \) and \( G \) according to the underlying state transition probabilities \( A_Q \) and \( A \). This is equivalent to performing a random walk on the product graph \( G_x = (V_x, E_x) \) [8], where:

\[
V_x = \{(q_i, v_j) : q_i \in V_Q \land v_j \in V\},
\]

\[
E_x = \{((q_i, v_j), (q_i', v_j')) : (q_i, q_i') \in E_Q \land (v_j, v_j') \in E\}.
\]

As discussed in [8] the transition probability matrix of the new Markov chain defined on the product graph \( G_x \) can be computed as \( A_X = A_Q \otimes A \), where \( \otimes \) is the Kronecker product of two matrices. Furthermore, we can compute the stationary distribution of the random walk on the product graph as \( \pi_x = \pi_Q \otimes \pi \), where \( \pi_Q \) and \( \pi \) respectively are the stationary distributions of the underlying Markov chains on \( G_Q \) and \( G \). The distributions \( \pi_Q \) and \( \pi \) correspond to the normalized left eigenvectors of the state transition probability matrices \( A_Q \) and \( A \). Now assume that the walker spends a random amount of time with mean \( \mu \), once it enters a given node \( v_{xh} = (q_i, v_j) \) in the product graph \( G_x \). If the mean time spent at the node \( v_{xh} \) corresponds to \( S_{ij} \), the sequence similarity score between the query protein \( q_i \in V_Q \) and target protein \( v_j \in V \), the long-run proportion of time that the walker spends at node \( v_{xh} = (q_i, v_j) \) can be computed as:

\[
c_{ij} = \pi_x S_{ij} = \frac{\pi_{xh} \mu h}{\sum_{h=1}^n \pi_{xh} \mu h} = \frac{\pi_Q S_{ij}}{\sum_{i=1}^n \pi_i S_{ij}}.
\]

Here, \( c_{ij} = \pi_x S_{ij} \) is the proportion of time that the semi-Markov random walker stays simultaneously at \( q_i \in V_Q \) and \( v_j \in V \). We use the score \( c_{ij} \) as the correspondence score between protein \( q_i \) in the query network and the protein \( v_j \) in the target network. For larger \( c_{ij} \), the random walker is more likely to stay in the corresponding pair nodes, hence it is more likely that the subnetwork region around those proteins will be more similar. This approach allows us to efficiently combine topological similarity and sequence similarity to estimate the node correspondence scores. These scores are used for iterative network reduction, as described in the next subsection.

It is noteworthy that similar random walk based scheme is proposed in [9] to obtain scores between nodes of two PPI networks. However, in [9], scores between two nodes in different PPI networks are obtained by linearly combining the topological similarity scores and the sequence similarity scores.

### 2.2. Iterative Network Reduction

To identify the subnetwork region in the target network that resembles the query network, we use an iterative scheme to reduce the search space in the target network. We start the iteration by using the complete target network \( G^{(0)} = G \). In each iteration, we reduce the network \( G^{(l-1)} \) to \( G^{(l)} \) as follows.

Consider the \( l \)-th iteration, where the current target network has been reduced to \( G^{(l-1)} = (V^{(l-1)}, E^{(l-1)}) \) during the previous iterations, where \( |V^{(l-1)}| = m_{l-1} \). We compute the correspondence score \( c_{ij} \) between every pair of nodes \( q_i \) (in the query) and \( v_j \) (in the reduced target network \( G^{(l-1)} \)). Since small correspondence scores indicate that the corresponding proteins have little similarity, such proteins can be removed from the target network. To this aim, for each protein \( v_j \) in the target network \( G^{(l-1)} \), we compute its best correspondence to the nodes in the query network:

\[
m_j = \max_{i=1:k} c_{ij}.
\]

Then we sort the nodes in the target network into an ordered set \( M = \{v_{(1)}, v_{(2)}, \ldots, v_{(m_{(l-1)})}\} \), such that \( m_{(1)} \leq m_{(2)} \leq \ldots \leq m_{(m_{(l-1)})} \). We pick the first \( r_l \) elements of \( M \) (which correspond to nodes that have minimal correspondence to nodes in the query network) and discard them from the target network to obtain \( G^{(l)} \). Thus, \( G^{(l)} \) is the induced graph over the remaining vertices, i.e., \( G^{(l)} = G^{(l-1)} \{V^{(l-1)} \setminus \{v_{(1)}, v_{(2)}, \ldots, v_{(r_l)}\}\} \). We repeat this process using the updated target network \( G^{(l)} \).

To summarize, in each iteration, we first update the correspondence scores between all pairs of proteins in the query network \( G_Q \) and the current target network \( G^{(l-1)} \). Then, we identify the set of proteins in \( G^{(l-1)} \) that have minimal correspondence to the proteins in the query network and remove them from the current target network \( G^{(l-1)} \) to obtain \( G^{(l)} \). The iteration stops when \( m_l < 2k \), namely, when we have reached a network whose size is less than twice the size of the query network. Otherwise, we repeat the iteration. In order to expedite the reduction process and reduce the false
positive rate, after each iteration, we also remove all nodes in the reduced network $G^{(l)}$ whose degree is zero (i.e., they do not have any neighbors), unless the node has the best correspondence score to one of the query nodes.

### 2.3. Insertions and Deletions

It is possible that some nodes in the query network may not have any matching nodes in the target network (referred as “deleted” nodes). On the other hand, there may be nodes in the identified subnetwork, that may not have matching nodes in the query network (referred as “inserted” nodes). Such inserted nodes are used to connect other matching nodes in the network.

The proposed network querying algorithm can naturally handle deleted nodes, since if a node is deleted while its two neighbors are still connected, the random walker will still have a good chance to walk through these nodes. In this way, their interaction pattern will still be considered by the algorithm. For inserted nodes, we apply a probabilistic scheme before starting the network reduction process by adding some auxiliary edges that can handle such insertions. In this scheme, we find nodes in the target network with zero degree but have non-zero average similarity with the nodes in the query network. For such a node $v_j$, we find set of query nodes $L_j$ with positive sequence level similarity, i.e., $L_j = \{ q_i \in V_Q \mid S_{ij} > 0 \}$.

Then, among the nodes in the target network that are similar to the neighbors of the query nodes in $L_j$, we pick two nodes randomly and with a probability of $p$, we add an edge between $v_j$ and those nodes. These edges can help dealing with insertions in the query network.

### 2.4. Matching Query nodes

When the iterations stop, we get a reduced target network $G' = (V', E')$ which has less than $2k$ nodes. As the final step, we update the correspondence scores $c_{ij}$ between the query nodes and the nodes in $V'$ using (2). Next, we consider an edge-weighted bipartite graph $\tilde{G} = (Q \cup V', E')$ in which each edge $e = (q_i, v'_j) \in E'$ is associated with the weight $c_{ij}$, the correspondence score between nodes $q_i$ in $G_Q$ and $v'_j$ in $G'$. On this bipartite graph, we employ the maximum weighted bipartite matching algorithm to match the nodes in the query network with those in $G'$. If we represent the set of matched nodes of $G'$ as $V_S$, the induced subnetwork $G'[V_S]$ will be the matched subnetwork of the target PPI network to the query network.

### 3. RESULTS

We conducted a set of experiments using both synthetic and real PPI networks to verify the accuracy and efficiency of the proposed network querying algorithm. In these experiments, we used the parameters $\tau_1 = \max\{0.05|V^{(l-1)}|, 1\}$ and $\rho = 0.01$.

In our first experiment, we generate a set of densely connected random query networks whose size ranges from $k = 5$ to $k = 30$. For each of these query networks, we generate a random target network with 2,000 vertices, in which the query network is embedded after randomly removing several edges, deleting 10% of the query nodes and inserting 15% of new nodes. As discussed in [1], most biological networks do not follow the Erdos-Renyi random network model, in which each pair of nodes are connected with constant probability. Rather, their structure resembles a scale-free topology, where the degree distribution follows a power law. Thus, we use the BA model [10] to generate a scale-free target network. In addition to this, we also assign random similarity scores between the query and target nodes. The similarity score between matching nodes in the two networks is randomly selected according to the Gaussian distribution $\mathcal{N}(200, 50)$, while other similarity scores are chosen from a different Gaussian distribution $\mathcal{N}(70, \sigma_2)$, where $\sigma_2$ varies between 0 and 100. We picked these numbers so that they match the conditions in real biological networks. As $\sigma_2$ increases, the overlap between two distributions increase, hence sequence similarity scores become less discriminative for querying purpose.

Percentage of correctly predicted nodes (sensitivity) is shown in Fig 2. As we can see, for all network sizes, we can recover the embedded target subnetwork with high accuracy even when the similarity scores are not very discriminative. This clearly reveals that the proposed querying algorithm effectively uses interaction patterns to accurately detect similar subnetwork regions.

In our second experiment, we use known protein complexes (set of highly connected proteins) of one species as our query to search the PPI network of another species, and analyze the biological validity of the retrieved subnetworks. We use PPI networks of fly (with 14,098 proteins and 25,830 interactions), worm (with 19,756 proteins and 4,572 interactions), and human (with 22,369 proteins and 36,386 interactions) as the target network, and we pick queries from yeast and worm. The networks used in this experiment and the sequence similarity scores (BLAST score) are obtained from [9].

The query examples are chosen from known complexes in the literature [11]. The querying results are summarized in Table 1. To evaluate the biological validity of the predicted matches, we use a software called GO TermFinder that computes the functional coherence of a set of proteins and its significance according to the Gene Ontology (GO) annotations. Table 1 shows the resulting p-values obtained by comparing the query and the retrieved target subnetwork. The small p-values indicate that the proposed scheme is sensitive. To measure the specificity of our algorithm, we also report the number of matches in the last column of Table 1. We counted the number of correctly matched proteins (out of total matching proteins) based on previously reported querying results reported in [11]. Although the results reported in [11] may not necessarily represent biological truth, significant overlap with these results imply that the specificity of the proposed querying algorithm is reasonably high. Fig. 3 shows one of the querying results (fifth example in Table 1) for a query network that corresponds to the protein catabolism process. Although the target network has some missing interactions, we can see that the algorithm can effectively predict the matching proteins with high accuracy. For all examples reported in Table 1, it took only about ten seconds for querying using Matlab on a personal computer.

![Fig. 2. Performance analysis on the synthetic network for query networks with different size. The sensitivity reports the percentage of correctly matched nodes.](image-url)
Table 1. Cross-species querying results.

<table>
<thead>
<tr>
<th>Query organism</th>
<th>Target organism</th>
<th>k</th>
<th>Biological Process</th>
<th>p-value</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yeast</td>
<td>fly</td>
<td>16</td>
<td>Peroxisome</td>
<td>3.50E-004</td>
</tr>
<tr>
<td>2</td>
<td>yeast</td>
<td>fly</td>
<td>5</td>
<td>Cell proliferation</td>
<td>6.30E-004</td>
</tr>
<tr>
<td>3</td>
<td>yeast</td>
<td>fly</td>
<td>24</td>
<td>RNA processing</td>
<td>3.05E-012</td>
</tr>
<tr>
<td>4</td>
<td>yeast</td>
<td>fly</td>
<td>14</td>
<td>Pheromone signal transduction</td>
<td>5.00E-013</td>
</tr>
<tr>
<td>5</td>
<td>yeast</td>
<td>worm</td>
<td>17</td>
<td>Protein catabolism</td>
<td>1.78E-015</td>
</tr>
<tr>
<td>6</td>
<td>worm</td>
<td>fly</td>
<td>10</td>
<td>Cell death</td>
<td>5.60E-004</td>
</tr>
<tr>
<td>7</td>
<td>yeast</td>
<td>human</td>
<td>7</td>
<td>Cell cycle</td>
<td>7.67E-006</td>
</tr>
</tbody>
</table>

Fig. 3. Result of querying a worm (C.elegans) PPI network using a query network that corresponds to a protein catabolism process in yeast (S.cerevisae) using the proposed algorithm (example 5 in Table 1). The query complex is obtained from [11]. Solid lines represent the interactions between proteins of a given species, while dashed lines connect matching nodes between the two species.

4. CONCLUSION

In this paper, we proposed a novel network querying algorithm for searching large-scale biological networks. In this approach, we used a semi-Markov random walk framework to compute the correspondence scores between nodes in the query and target networks. These scores can effectively capture topological similarity of the subnetworks around the nodes as well as their sequence similarity. These scores were used to iteratively reduce the search space in the target network by discarding the nodes with minimal correspondence to the query nodes. Unlike most of the previous network querying algorithms, our algorithm has only a polynomial computational complexity in terms of the size of both networks. Furthermore, it can be efficiently implemented by exploiting sparsity in real biological networks. Experimental results based on synthetic and real networks clearly demonstrate the accuracy and efficiency of the proposed algorithm.

5. REFERENCES